VII International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2017
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

# NUMERICAL SIMULATION OF SHOCK-TUBE PISTON PROBLEMS WITH ADAPTIVE, ANISOTROPIC MESHES

**BARBARA RE*, CÉCILE DOBRZYNSKI† AND ALBERTO GUARDONE***

*Department of Aerospace Science & Technology
Politecnico di Milano, Via La Masa 34, 20146 Milano, Italy
e-mail: alberto.guardone@polimi.it, web page: crealab.polimi.it

†Bordeaux INP, Enseirb-Matmca
Institut de Mathmatiques de Bordeaux
200 av de la vieille Tour, 33405 Talence cedex, France

**Key words:** Finite-Volume schemes, compressible inviscid flows, shock-tube, anisotropic mesh adaptation

**Abstract.** Numerical simulations of the flow generated inside a shock-tube by the motion of a magnetically- driven piston are carried out using a novel finite volume adaptive scheme for dynamic meshes. Local modifications of the grid topology, including the addition or deletion of grid nodes are interpreted as a series of fictitious, continuous deformations of the mesh, thus allowing mesh adaptation to be described within the Arbitrary Lagrangian Eulerian (ALE) framework. The local deformations of the mesh elements are taken into account in a conservative fashion by adding additional fictitious fluxes to the ALE formulation of the governing equations for inviscid compressible flows. The solution on the new grid is recovered without any explicit interpolation. Therefore, the method automatically guarantees the solution to be conservative by construction. This peculiar capability is here exploited in preliminary Fluid-Structure Interaction (FSI) computations of compressible shocked flows with rigid, moving bodies. Anisotropic mesh adaptation is used to improve the computational efficiency. The solution compares fairly well with the analytical one-dimensional model.

## 1 INTRODUCTION

In unsteady numerical simulations, it is often required to deform the grid to comply with the motion experienced by the boundary of the computational domain. In particular, if the interaction between the flow and a deformable structure is investigated as in Fluid-Structure Interaction (FSI) problems, the capability to accurately accomplish this task becomes of primary importance [1, 2, 3].

To solve flow governing equations on a moving or deforming mesh, the formulation of the governing equations has to be modified in some way to account for the relative motion of the grid with respect to the fluid. Arbitrary Lagrangian-Eulerian (ALE) approach is widely used to this aim, since it allows to solve the equations on control volumes that can move and deform independently from the local fluid velocity [4, 5]. A second aspect that should be taken into consideration concerns the movement of the interior mesh points, consequent to the motion prescribed on the boundaries. Small deformations can be easily tackled by means of dynamic grid methods [6] in which the grid connectivity is kept fixed and the grid nodes are simply moved, in order to distribute the boundary deformation among internal grid elements according on their shape and volume. However, the fixed-topology constraint limits the displacement that the mesh is capable to handle without generating entangled or poor-quality elements that may jeopardize the numerical solution of the flow problem.

Mesh adaptation techniques can be exploited along with dynamic grid methods to deal with large boundary displacement [7, 8]. When the mesh quality falls below a certain threshold or null-volume elements appear, the computational grid is modified by local connectivity changes, such as edge swapping, node insertion or deletion [9, 10, 11, 12]. Then, the solution is transfered from the old grid to the adapted one, usually by means of an interpolation step. Unfortunately, this operation may prevent the solution accuracy and undermine the stability and conservativeness of the numerical scheme.

A different approach has been proposed by Guardone and co-workers [13, 14] that exploits mesh adaptation capabilities within the ALE framework, so that it is able to cope with large deformations and at the same time avoids any explicit interpolation step. The volume changes due to mesh adaptation are described as a series of fictitious continuous deformations of the finite volumes that compose the computational domain and they are included into the governing equations as additional ALE fluxes. In this ways, the properties of the fixed-connectivity ALE scheme, such as stability and conservativeness, are preserved also over adaptive grids.

In the present paper, the bi-dimensional version of this scheme is exploited to compute the flow field generated in a shock-tube by a magnetically driven piston. The size, the shape and the orientation of the grid elements is controlled through anisotropic mesh adaptation [15, 16, 17]. Indeed, the mesh is locally modified according to a metric tensor field based on the Hessian of the numerical solution. Due to the one-dimensional character of the flow field, the possibility of prescribing different grid spacing in different direction is crucial and, with respect to isotropic adaptation, it leads to a lower number of grid nodes.

## 2 NUMERICAL METHOD

The ALE formulation of the Euler equations for a control volume $\mathcal{C}(t)$ moving at velocity $\boldsymbol{v}(\boldsymbol{x}, t)$ reads

$$\frac{d}{dt}\int_{\mathcal{C}(t)} \mathsf{u}\, d\boldsymbol{x} + \oint_{\partial\mathcal{C}(t)} [\mathbf{f}(\mathsf{u}) - \mathsf{u}\boldsymbol{v}] \cdot \boldsymbol{n}\, ds = 0\,, \qquad \text{with} \quad \mathbf{f}(\mathsf{u}) = \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{m}\otimes\boldsymbol{m}/\rho + P(\mathsf{u})\mathbb{I}^3 \\ \left[E^T + P(\mathsf{u})\right]\boldsymbol{m}/\rho \end{bmatrix} \tag{1}$$

where $\mathsf{u} = [\rho,\, \boldsymbol{m},\, E^T]^{\mathrm{T}}$ is the vector of conservative variables (density, momentum density, and total energy density, respectively), $\mathbf{f}(\mathsf{u})$ is the inviscid flux function, $\boldsymbol{n}(\boldsymbol{x}, t)$ denotes the outward unit vector normal to the cell boundary $\partial\mathcal{C}$, $P$ is the pressure function (which depends on the adopted thermodynamic models) and $\mathbb{I}^2$ is the $2 \times 2$ identity matrix. Suitable initial and boundary conditions have be to specified to complement the previous equation [18].

When solving flow equations on a moving computational domain, the evaluation of the geometric quantities connected to the grid movement is crucial. In this regard, it is usually recognized that the fulfillment of the so-called Geometric Conservation Law (GCL) positively affects the stability and the accuracy of the numerical scheme for dynamic grids and allows larger time steps with respect to non-compliant schemes [19, 20, 22, 21]. For the governing equations (1), the GCL can be expressed as

$$\frac{d}{dt}\int_{\mathcal{C}(t)} d\boldsymbol{x} = \oint_{\partial\mathcal{C}(t)} \boldsymbol{v}\cdot\boldsymbol{n}\, ds\,. \tag{2}$$

To spatially discretize Eq. (1), the computational domain $\Omega$ is split in $N_V$ non-overlapping finite volumes $\mathcal{C}_i$ and a suitable approximation $\phi$ is used to integrate the flux that each finite volume exchanges with its neighbors. In the present work, a high-resolution scheme based on the Total Variation Diminishing (TVD) approach [23] is used for the integrated numerical domain fluxes $\phi$, by exploiting a flux limiter to control the switch from the centered second-order approximation scheme to the Roe scheme near discontinuities. If node $i$ lies on the boundary, a similar approximation $\phi^\partial$ is introduced also for the integrated numerical flux across the portion of the cell interface that lies on the boundary, i.e. $\partial\mathcal{C}_{i,\partial} = \partial\mathcal{C}_i \cap \partial\Omega$. The spatially discrete Euler equations read

$$\frac{d}{dt}\left[V_i \mathsf{u}_i\right] = \sum_{k\in\mathcal{K}_{i,\neq}} \phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \boldsymbol{\eta}_{ik}) + \phi^\partial(\mathsf{u}_i, \nu_i, \boldsymbol{\xi}_i), \qquad i \in \mathcal{K} \tag{3}$$

where $\mathsf{u}_i$ is the average value of $\mathsf{u}$ over $\mathcal{C}_i$ and $\mathcal{K}$ is set of the grid nodes. We have defined also the set of the neighboring nodes of the finite volume $i$ as $\mathcal{K}_{i,\neq} = \{k \in \mathcal{K} \mid \partial\mathcal{C}_{ik} \neq \emptyset\}$, where $\partial\mathcal{C}_{ik} = \partial\mathcal{C}_i \cap \partial\mathcal{C}_k$ indicates the portion of the cell interface shared by the volumes $i$ and $k$. Moreover, we have introduced the integrated normals $\boldsymbol{\eta}_{ik}$, $\boldsymbol{\xi}_i$ and the interface velocities $\nu_{ik}$ and $\nu_i$, defined as

$$\boldsymbol{\eta}_{ik} = \int_{\partial\mathcal{C}_{ik}} \boldsymbol{n}_i\,, \qquad \boldsymbol{\xi}_i = \int_{\partial\mathcal{C}_{i,\partial}} \boldsymbol{n}_i\,, \qquad \nu_{ik} = \int_{\partial\mathcal{C}_{ik}} \boldsymbol{v}\cdot\boldsymbol{n}_i\,, \qquad \nu_i = \int_{\partial\mathcal{C}_{i,\partial}} \boldsymbol{v}\cdot\boldsymbol{n}_i\,. \tag{4}$$

Equation (3) is then integrated in time by means of a Backward Euler scheme, namely

$$\frac{V_i^{n+1}\mathsf{u}_i^{n+1} - V_i^n \mathsf{u}_i^n}{\Delta t} = \sum_{k \in \mathcal{K}_{i,\neq}} \phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \boldsymbol{\eta}_{ik})^{n+1} + \phi^{\partial}(\mathsf{u}_i, \nu_i, \boldsymbol{\xi}_i)^{n+1} \,. \tag{5}$$

The previous non-linear system is solved through a pseudo-time step technique [24], in which, according to the defect-correction approach [25], a modified Newton method is used to solve the linear system at each pseudo time step.

Similarly, we obtain the Discrete Geometric Conservation Law (DGCL) by applying the same spatial and temporal discretization to Eq. (2):

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} = \sum_{k \in \mathcal{K}_{i,\neq}} \int_{\partial \mathcal{C}_{ik}} \boldsymbol{v} \cdot \boldsymbol{n}_i + \int_{\partial \mathcal{C}_{i,\partial}} \boldsymbol{v} \cdot \boldsymbol{n}_i = \sum_{k \in \mathcal{K}_{i,\neq}} \nu_{ik}(t) + \nu_i(t) \tag{6}$$

The constraint expressed by the previous relation can be enforced by splitting the contributions pertaining to the domain and to the boundary, as

$$\Delta V_{ik}^{n+1} = \Delta t \, \nu_{ik}^{n+1} \qquad \text{and} \qquad \Delta V_{i,\partial}^{n+1} = \Delta t \, \nu_i^{n+1} \,, \tag{7}$$

where $\Delta V_{ik}^{n+1}$ and $\Delta V_{i,\partial}^{n+1}$ are the volume swept during the interval from $t^n$ to $t^{n+1}$ by the interface portions $\partial \mathcal{C}_{ik}$ and $\partial \mathcal{C}_{i,\partial}$, respectively. Since we consider known the grid motion, relations (7) allow to easily compute GCL-compliant interface velocities from the positions of the grid nodes at the beginning and at the end of the time step if the finite volumes experience a continuous deformation.

In [13, 14], it has been described how it is possible to compute the swept volumes $\Delta V_{ik}^{n+1}$ and $\Delta V_{i,\partial}^{n+1}$ also if the local connectivity changes. A three-steps series of fictitious collapse and expansion operations can be exploited to describe node insertion, node deletion and edge swap as a series of continuous deformations of the finite volumes involved in the connectivity change. In particular, the grid is modified when all involved cell interfaces have a null area, so that this operation does not affect the fulfillment of the DGCL. In this way, we are able to compute the interface velocities through Eqs. (7) also when local grid topology is modified and to include the volume changes due to mesh adaptation into the standard ALE scheme, i.e Eq. (1), without undermining the fixed-connectivity properties and enforcing conservativeness. The main points of this procedure are described in Fig. 1 for the node insertion through edge split. A similar procedure can be applied also to different adaptation techniques [14, 26].

### 2.1 Anisotropic mesh adaptation

Mesh adaptation is widely used in CFD to cope with large boundary displacements but also to increase solution accuracy, provided that a proper estimation of the solution error is available. Indeed, thanks to a posteriori analysis, the error due to the numerical approximation can be related to the grid spacing and, in particular, an upper-bound of

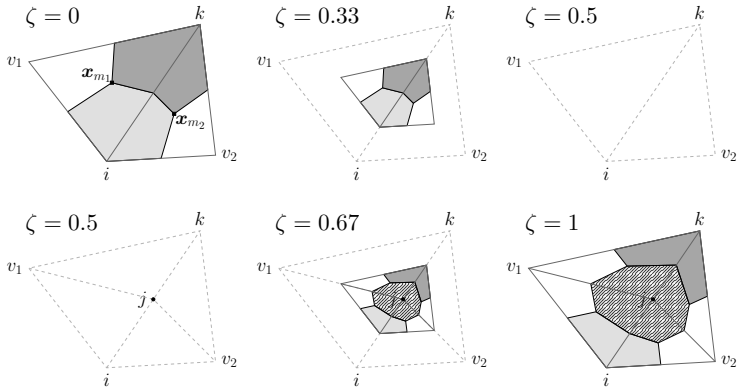


**Figure 1**: Three-steps procedure applied to the split of edge $e_{ik}$. The non-dimensional time $0 \leq \zeta \leq 1$ is used to describe the different fictitious steps. The dashed grey lines show the grid connectivity in the original/final configuration ($\zeta = 0/\zeta = 1$), while the portions of the finite volumes associated to $i$, $k$ and $j$ are shown with light grey □, dark grey ■ and the pattern ▨ , respectively. The label $\boldsymbol{x}_{m_i}$ indicates the barycenter of the element $m_i$. In the first row the collapse phase $0 < \zeta < 0.5$ is depicted: the quadrilateral $i$-$k$-$v_1$-$v_2$, composed by the elements that share the edge $e_{ik}$ at $\zeta = 0$, is collapsed over its mid-point. When it reaches a null area, the connectivity is changed ($\zeta = 0.5$): the new point $j$ is inserted, the edge $e_{ik}$ is split into two edges ($i$-$j$ and $k$-$j$) and two new edges are created to connect $j$ to $v_1$ and $v_2$. The second row displays the expansion procedure $0.5 < \zeta < 1$: the nodes $i$, $k$, $v_1$, $v_2$ return to their original positions to reach the final configuration (at $\zeta = 1$).

the interpolation error can be built on the basis of the Hessian matrix $\mathcal{H}_u$ of a certain solution variable $u$ [27, 17].

The target grid spacing can be described through a suitable metric field $\mathcal{M}(\boldsymbol{x})$ that prescribes the size, the shape and the orientation of the mesh elements [28, 29, 30]. For a bi-dimensional domain $\Omega$, the metric field $\mathcal{M}(\boldsymbol{x})$ is a field of symmetric positive matrices of $\mathbb{R}^{2\times 2}$, that defines a Riemannian structure over $\Omega$ [31]. The length of a vector $\boldsymbol{w}$ in terms of the anisotropic map $\mathcal{M}$ is given by

$$\|\boldsymbol{w}\|_{\mathcal{M}} = \sqrt{\boldsymbol{w}^{\mathrm{T}} \mathcal{M} \boldsymbol{w}} \, . \tag{8}$$

and, given a point $O$, the locus of the points $P$ such that $\|P - O\|_{\mathcal{M}} = h$ is represented by an ellipsoid, centered at $O$. The shape of the element is associated to the lengths of the semi-axes of the ellipse, which are defined by the eigenvalues of $\mathcal{M}$, while the orientation (of the ellipse and of the element) is defined by its eigenvectors.

Following [17], it is possible to control the error over the mesh by defining at each grid point $\boldsymbol{x}_i$ the metric map $\mathcal{M} = R\widetilde{\Lambda}L$ with

$$\widetilde{\Lambda} = \mathsf{diag}\left\{ \min\left( \max\left( \frac{c|\lambda_p|}{\epsilon}, \frac{1}{\ell^2_{\max}} \right), \frac{1}{\ell^2_{\min}} \right) \right\} ,$$

where $\lambda_p$, $R$ and $L$ are respectively the $p$-th eigenvalue, the matrix of the right and of the left eigenvectors of the Hessian matrix $\mathcal{H}_u(\boldsymbol{x}_i)$, $\epsilon$ is a given threshold for the maximum

tolerate error, $c$ is a constant that for bi-dimensional problems is 2/9 and $\ell_{\max}/\ell_{\min}$ are the maximum/minimum edge length. Then, mesh adaptation techniques can be used to equi-distribute this error over the mesh by locally modifying the element size [27, 17], since according to the equi-distribution principle, the minimum solution error is achieved over a *unit mesh*, i.e. a mesh that has all edges of unit length according to Relation (8) [28].

## 3 Computational procedure

In the present work, the external re-mesher library is exploited to efficiently modify the computational grid [32]. Figure 2 outlines the whole computational procedure for the unsteady simulations of the magnetically-induced piston motion. At the beginning of the time step $t^n \leq t \leq t^{n+1}$, the grid and the solution are respectively label as $\mathcal{K}^n$ and $\mathsf{u}(t^n, \mathcal{K}^n)$. Then, the following operations are performed:

1. Mesh deformation: the position of the piston is computed given the flow pressure and the prescribed force $\mathcal{F}_P(t^n)$, the internal grid nodes are displaced to cope with the boundary movement, as described in [26]. The new grid is labeled $\mathcal{K}^{n+}$.
2. Prediction: the solution at next time step, over the grid $\mathcal{K}^{n+}$, is computed. This step prevents a delay between the solution-based mesh adaptation and the actual geometry.
3. Error estimation: the metric map $\mathcal{M}$ is computed on the basis of the new solution $\mathsf{u}^{n+} = \mathsf{u}(t^{n+1}, \mathcal{K}^{n+})$.
4. Mesh adaptation: the grid $\mathcal{K}^{n+}$ and the metric map $\mathcal{M}(\mathsf{u}^{n+})$ are passed as inputs to the library mmg2d, which locally adapt the grid.
5. Mesh update: the flow solver receives from mmg2d all performed local modifications $\Delta\mathcal{K}^{n+}$, that are needed to update the finite volumes discretization and to compute the swept volumes $\Delta V$ by means of the three-steps procedure [14].
6. Computation of the solution $\mathsf{u}^{n+1}$ on the grid $\mathcal{K}^{n+1}$, using as initial guess $\mathsf{u}^{n+}$.

## 4 RESULTS

The described method is exploited to simulate the motion of a magnetically-driven piston, controlled by a prescribed force. First of all, we have estimated this force by imposing an harmonic motion to the piston and by evaluating the generated pressure field. Hence, a preliminary simulation is carried out in which we enforce the position of the piston as $x(t) = x_0 + A\cos(2\pi\, f\, t)$, with an amplitude $A = -0.5$ m and a frequency $f = 30$ Hz. The maximum speed reached by the piston during this motion is $V_{P_{\max}} = 94.25$ m/s, which in the case of an impulsively start would generate a shock wave traveling with a Mach number $M_s = 1.18$. In the present simulation, the piston is not started impulsively, so no shock waves are generated at the beginning of the motion.

The length and the height of the piston are $L_P = 0.1$ m and $h_P = 0.05$ m, respectively, and initially it is filled with still air at $T_0 = 288.15$ K and $P_0 = 1$ atm. The aim of this preliminary simulation is to reproduce an infinite-length shock-tube, so the computational domain consists in 20 m-long tube, with open ends at both sides, and the piston is placed

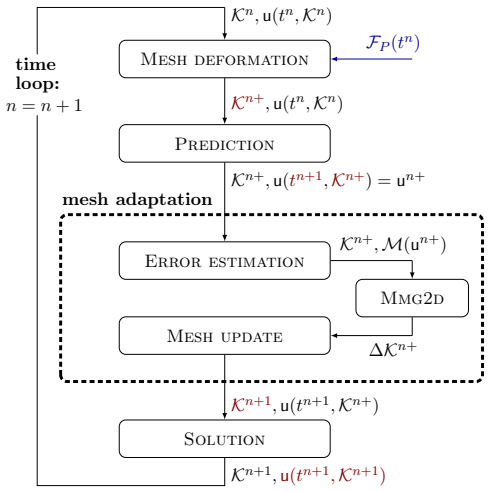


**Figure 2**: Adaptive computational procedure for unsteady problems. The grid $\mathcal{K}^{n+}$ complies with the boundary displacement at $t^{n+1}$ and, over it, the solution $\mathsf{u}^{n+}$ is computed in the *Prediction* step. The metric field $\mathcal{M}(\mathsf{u}^{n+})$ is passed as input to mmg2d which communicates to the flow solver all performed modifications $\Delta\mathcal{K}^{n+}$, so that it can compute the swept volumes $\Delta V$ due to mesh adaptation. Finally, the solution at $t^{n+1}$ over the adapted grid $\mathcal{K}^{n+1}$ is computed.

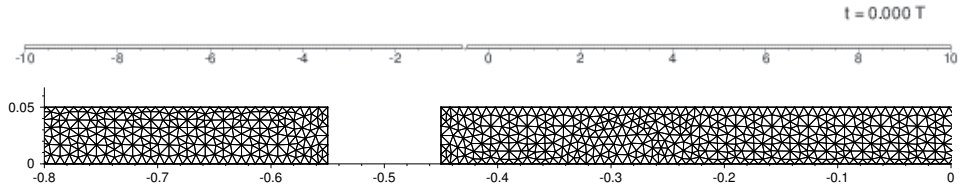


**Figure 3**: Computational grid for preliminary simulation of the piston subject to an harmonic motion. Top: whole domain; Bottom: detail of the mesh near the piston, with at the beginning of the simulation is centered at $x = -0.5$ m.

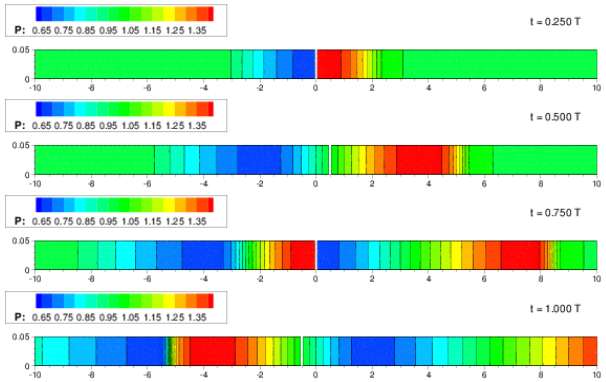


**Figure 4**: Pressure contour plots of the piston during the first period of the harmonic motion. The pressure is made dimensionless with respect to the initial value $P_0$. A different aspect ratio on the $x$ and $y$ axis is used to make the results clearer.

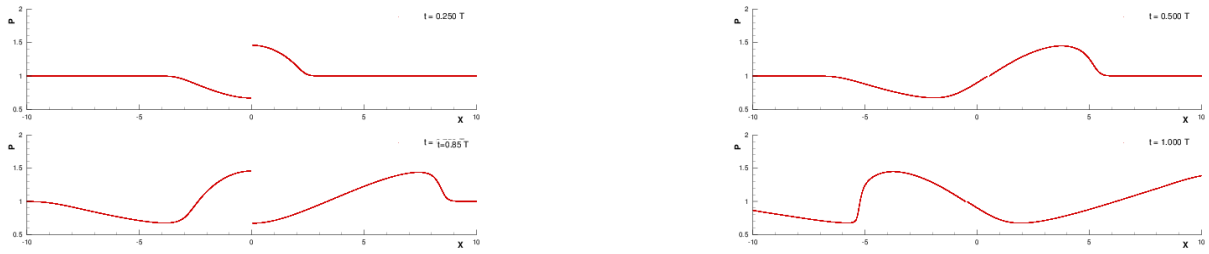


**Figure 5**: Pressure at all grid points during the first period of the harmonic motion of the piston. The pressure is made dimensionless with respect to the initial value $P_0$.

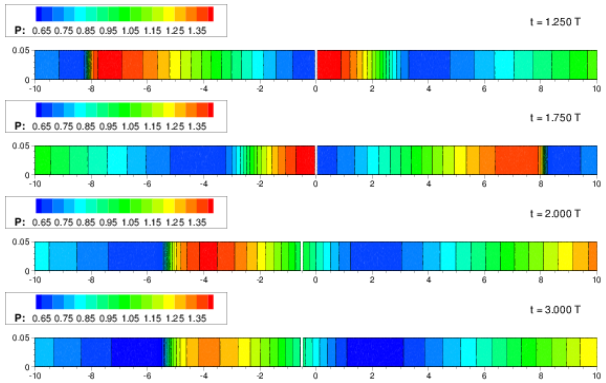


**Figure 6**: Pressure contour plots of the piston during the second and third period of the harmonic motion. The pressure is made dimensionless with respect to the initial value $P_0$. A different aspect ratio on the $x$ and $y$ axis is used to make the results clearer.

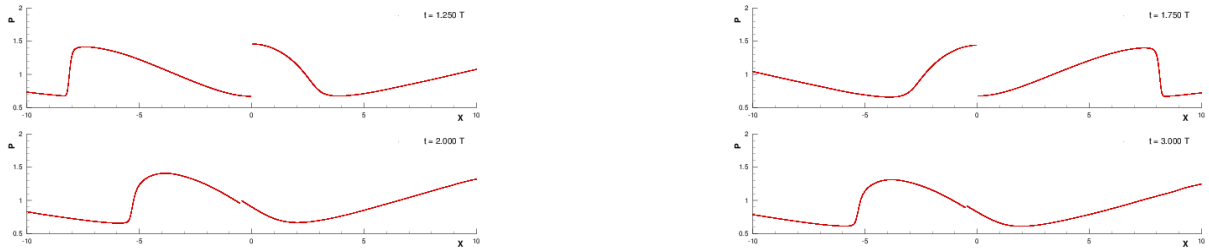


**Figure 7**: Pressure at all grid points during the second and third period of the harmonic motion of the piston. The pressure is made dimensionless with respect to the initial value $P_0$.

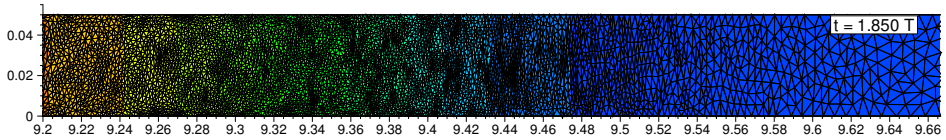


**Figure 8**: Detail of the adapted mesh during the preliminary simulation of the piston with harmonic motion.

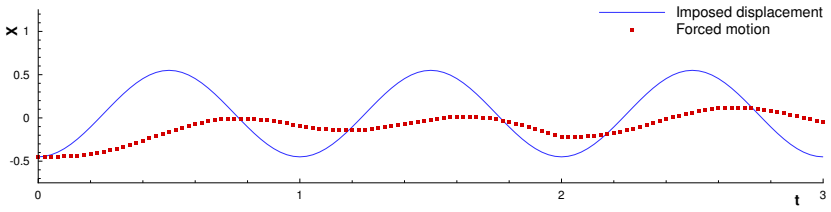


**Figure 9**: Position of the piston during the harmonic motion (Imposed displacement) and the final simulation in which the force is prescribed (Force motion).

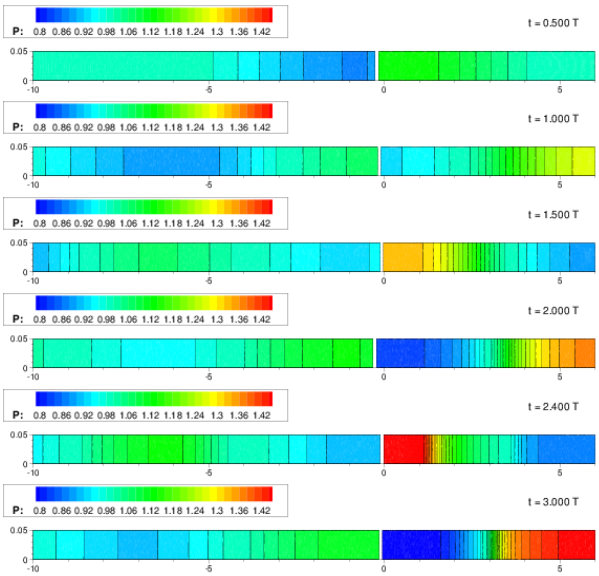


**Figure 10**: Pressure contour plots of the simulation of the piston with a prescribed force. The pressure is made dimensionless with respect to the initial value $P_0$. A different aspect ratio on the $x$ and $y$ axis is used to make the results clearer.

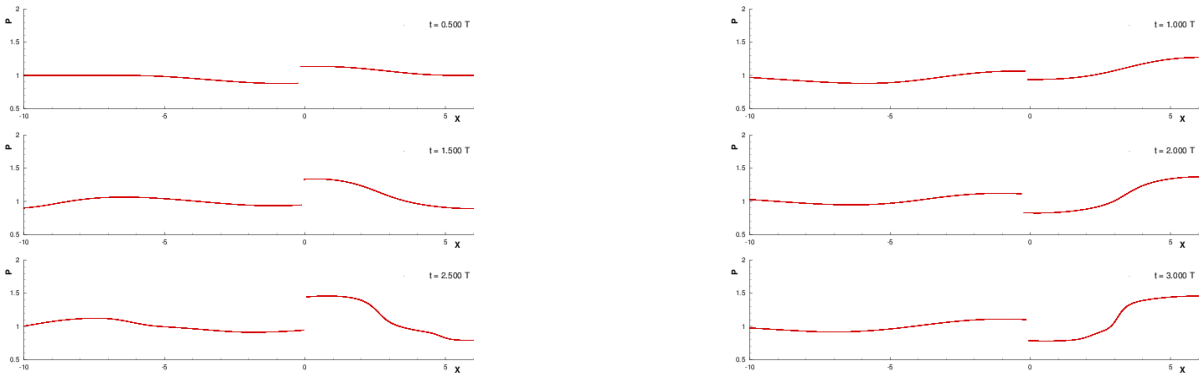


**Figure 11**: Pressure at all grid points during the simulation of the piston with a prescribed force. The pressure is made dimensionless with respect to the initial value $P_0$.

in the middle of the tube. The initial grid, shown in Fig. 3, is composed by 20 863 nodes and 36 724 triangles. Each period $T$ is divided in 400 time steps, so that the Courant number is approximately equal to the unity on the initial grid, characterized by a spacing of $\Delta x = 0.008$ m.

When the motion starts, expansive and compressive waves are generated at the opposite sides of the piston and they travel towards the open ends. Initially the fluid on the right side (positive $x$-direction) is compressed and the compressive waves generated during the time interval $0 < t < 0.25T$ are getting stronger and faster. However, since the piston velocity in this interval is quite small, no shock waves are generated. In the same interval, on the left side of the shock-tube, an expansion fan originates. For time $0.25T < t < 0.5T$, the piston speed is reduced, so that an expansion takes place on the right part and a compression on the left part. These phenomena continue also in the next quarter of period, although the direction of the piston is now towards negative $x$. At $t = 0.75T$, the velocity of the harmonic motion is maximum in magnitude and it decreases leading to an expansion on the left side and a compression on the right side. The compressive waves generated on the left side of the piston coalesce together generating a shock far from the piston side at about $t > 1.15T$. Figs 4 and 5 show the pressure field during this first period.

In the second period, the flow field is different from the first one, since the fluid in the piston has now a non-null velocity. In particular, a shock wave is generated also on the right side approximately at $t = 1.5T$, when the compressive waves generated at $0.75T < t < T$ coalesce. It should be remarked that, after the shock, the total pressure is lower, even if the shock intensity is quite small. Hence, also the flow field in the third period is slightly different, despite qualitatively the same phenomena take place. Figs 6 and 7 display the pressure field at some times for $T < t < 2T$. At the end of this preliminary simulation, the force due to the pressure acting on the sides of the piston is computed. During this simulation, the Hessian of the pressure is used to increase solution accuracy by means of anisotropic adaptation. Fig. 8 illustrates a detail of the mesh at $t = 1.85T$ near the shock propagating in the right part of the shock-tube.

The force computed in the preliminary simulation is now used to move the piston. In this new simulation, the shock-tube is represented by a wall on the right end and an open-end at left. Assuming a mass $m = 1.0$ kg, at each time step the velocity (in the $x$ direction) of the piston is computed as

$$V_P^{n+1} = V_P^n + \left( \mathcal{F}_P^n + \int_{\partial\Omega_P} P(\mathsf{u}^n)\, \boldsymbol{n}_x \, dS \right) / \left( m\, \Delta t \right) , \tag{9}$$

where $\mathcal{F}_P$ is the force used to move the piston, $P(\mathsf{u})$ is the pressure and $\boldsymbol{n}_x$ is the $x$-component of the normal to the piston surface, labeled $\partial\Omega_P$.

As expected, the motion prescribed in this second simulation is different, as shown in Fig. 9 because of the different flow fields. Moreover, the compressive waves generated by the piston motion are reflected back from the wall at approximately $t = 0.65T$. The

interaction of the reflected waves with the approaching ones modifies significantly the pressure field, which reaches also a higher maximum value, as shown in Figs. 10 and 11.

## 5 CONCLUSIONS

A novel adaptive scheme for the finite volume discretization of the Euler equations for unsteady bi-dimensional flow problems is used to exploit anisotropic mesh adaptation within the Arbitrary Lagrangian Eulerian framework, so that also the Geometric Conservation Law is automatically fulfilled and no interpolation of the solution is required between different grids. These features make this technique well suited to tackle Fluid Structure Interaction problems in which the conservativeness of the numerical scheme plays a crucial role.

The described numerical method is used to simulate a magnetically-driven piston in a shock-tube, subject to the prescribed force law that, in an open-ends tube corresponds to an harmonic motion. All significant flow features are correctly captured thanks to mesh adaptation.

## REFERENCES

[1] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, *Comput. Methods Appl. Mech. Engrg.* (1996) **134**:71–90.

[2] C. Farhat, M. Lesoinne, P. LeTallec, Load and motion transfer algorithms for fluid/ structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity, *Comput. Methods Appl. Mech. Engrg.* (1998) **157**:95–114.

[3] A. Masud, M. Bhanabhagvanwala, R. A. Khurram, An adaptive mesh rezoning scheme for moving boundary flows and fluidstructure interaction, *Comput. Fluids* (2007) **36**:77–91.

[4] C. Hirt, A. A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, *J. Comput. Phys.* (1974) **14** (3):227–253.

[5] J. Donea, S. Giuliani, J. Halleux, An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions, *Comput. Methods Appl. Mech. Engrg.* (1982) **33** (1):689–723.

[6] J. T. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, *AIAA journal* (1990) **28** (8):1381–1388.

[7] T. J. Baker, Mesh Movement and Metamorphosis, *Eng. Comput.* (2002) **18**:188-198.

[8] G. Compère, J. J. Jean-François Remacle, J. Hoffman, A mesh adaptation framework for dealing with large deforming meshes, *Int. J. Numer. Meth. Eng.* (2010) **82**:843–867.

[9] R. Löhner, Mesh Adaptation in fluid Mechanics, *Eng. Fract. Mech.* (1995) **50** (5/6):819–847.

[10] T. J. Baker, Mesh adaptation strategies for problems in fluid dynamics, *Finite Elem. Anal. Des.* (1997) **25** (3-4):243–273.

[11] C. Dobrzynski, P. Frey, Anisotropic Delaunay mesh adaptation for unsteady simulations, in: *Proceedings of the 17th international Meshing Roundtable*. Springer, (2008), pp. 177–194.

[12] C. Dapogny, C. Dobrzynski, P. Frey, Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, *J. Comput. Phys.* (2014) **262**:358–378.

[13] A. Guardone, D. Isola, G. Quaranta, Arbitrary Lagrangian Eulerian formulation for two-dimensional flows using dynamic meshes with edge swapping, *J. Comput. Phys.* (2011) **230** (20):7706–7722.

[14] D. Isola, A. Guardone, G. Quaranta, Finite-volume solution of two-dimensional compressible flows over dynamic adaptive grids, *J. Comput. Phys.* (2015) **285**:1–23.

[15] M.J. Castro-Díaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructured mesh adaption for flow simulations, *Int J Numer Meth Fluids* (1997) **25**:475–491.

[16] W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, M.-G. Vallet, Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles, *Int. J. Numer. Meth. Fluids* (2000) **32**:725–744.

[17] P.J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, *Comput. Methods Appl. Mech. Engrg.* (2005) **194**:5068–5082.

[18] R. J. LeVeque, Numerical methods for conservation laws, Birkhäuser, Basel, 1992.

[19] C. Farhat, P. Geuzaine, C. Grandmont, The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids, *J. Comput. Phys.* (2001) **174**:669–694.

[20] L. Formaggia, F. Nobile, Stability analysis of second-order time accurate schemes for ALEFEM, *Comput. Methods Appl. Mech. Engrg.* (2004) **193**(39-41):4097–4116.

[21] S. Étienne, A. Garon, D. Pelletier, Perspective on the geometric conservation law and finite element methods for ALE simulations of incompressible flow, *J. Comput. Phys.* (2009) **228** (7):2313–2333.

[22] D. J. Mavriplis, Z. Yang, Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes, *J. Comput. Phys.* (2006) **213**:557–573.

[23] A. Harten, High Resolution Schemes for Hyperbolic Conservation Laws, *J. Comput. Phys.* (1997) **135**:260–278.

[24] A. Jameson, Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings, *AIAA paper* (1991) **1596**:1–13.

[25] B. Koren, Defect correction and multigrid for an efficient and accurate computation of airfoil flows, *J. Comput. Phys.* (1988) **77**:183–206.

[26] B. Re, A. Guardone, C. Dobrzynski, An Adaptive Conservative ALE Approach to Deal with Large Boundary Displacements in Three-Dimensional Inviscid Simulations, in: *AIAA SciTech Forum*, Texas, 2017.

[27] V. Dolejší. Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes, *Computing and Visualization in Science* (1998) **1**(3):165–178.

[28] H. Borouchaki, P.-L. George, F. Hecht, P. Laug, E. Saltel, Delaunay mesh generation governed by metric specifications. Part I. Algorithms. *Finite Elem. Anal. Des.* (1997) **25**:61–83.

[29] S. Del Pino. Metric-based mesh adaptation for 2D Lagrangian compressible flows. *J. Comput. Phys.* (2011) **230**:1793–1821.

[30] T. Coupez. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *J. Comput. Phys.* (2011) **230**:2391–2405.

[31] P. Frey and P.-L. George. Mesh Generation: Application to Finite Element, Vol. 32, *Wiley* (2010).

[32] C. Dobrzynski, C. Dapogny, P. Frey, A. Froehly, www.mmgtools.org Mmg Platform.